

Optimizer algorithms and convolutional neural networks for text classification

Mohammed Qorich, Rajae El Ouazzani

Image Laboratory, ISNET Team, School of Technology, Moulay Ismail University of Meknes, Meknes, Morocco

Article Info

Article history:

Received Dec 2, 2022

Revised Feb 3, 2023

Accepted Mar 10, 2023

Keywords:

Convolutional neural network
Deep learning
Natural language processing
Optimization algorithms
Sentiment analysis
Text classification

ABSTRACT

Lately, deep learning has improved the algorithms and the architectures of several natural language processing (NLP) tasks. In spite of that, the performance of any deep learning model is widely impacted by the used optimizer algorithm; which allows updating the model parameters, finding the optimal weights, and minimizing the value of the loss function. Thus, this paper proposes a new convolutional neural network (CNN) architecture for text classification (TC) and sentiment analysis and uses it with various optimizer algorithms in the literature. Actually, in NLP, and particularly for sentiment classification concerns, the need for more empirical experiments increases the probability of selecting the pertinent optimizer. Hence, we have evaluated various optimizers on three types of text review datasets: small, medium, and large. Thereby, we examined the optimizers regarding the data amount and we have implemented our CNN model on three different sentiment analysis datasets so as to binary label text reviews. The experimental results illustrate that the adaptive optimization algorithms Adam and root mean square propagation (RMSprop) have surpassed the other optimizers. Moreover, our best CNN model which employed the RMSprop optimizer has achieved 90.48% accuracy and surpassed the state-of-the-art CNN models for binary sentiment classification problems.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Mohammed Qorich

Image Laboratory, ISNET Team, High School of Technology, Moulay Ismail University of Meknes
Meknes, Morocco

Email: mohamedqorich@gmail.com

1. INTRODUCTION

Recently, text classification (TC) attends a crucial interest in the natural language processing (NLP) field in light of the upgrading in deep learning research [1]. Actually, TC is the task of extracting labels from a given text data based on features selection [2] and it is used in numerous applications such as spam detection, topic labeling, question answering, and sentiment analysis [1]. The latter designates the task of identifying the polarity from reviews and opinions text either by multiple or binary classification [3]. Using deep learning algorithms, many researchers propose different methods and architectures to highly increase the performances in TC and sentiment analysis problems. Pal *et al.* [4] and Chamekh *et al.* [5] have adopted recurrent neural networks (RNN) models and long short-term memory (LSTM). Meanwhile, Sachin *et al.* [6] and Zulqarnain *et al.* [7] have employed gated recurrent units (GRU). Besides, Kim *et al.* [8] and Feng *et al.* [9] have implemented convolutional neural network (CNN) models, while Jain *et al.* [10] and Rehman *et al.* [11] have proposed hybrid models using CNN and RNN layers.

Despite the efforts made in these topics, these problems need more experimental aspects. In practice, the efficiency of a deep learning model not only relies on the used architectures, layers and activation functions, but also on the selection of the appropriate optimizers [12]. In effect, the choice of an optimizer almost stands

on best practices, online recommendations or even on random selections, and not relies on an empirical evidence approach due to the insufficiency of experiments [12]. Indeed, we proffer in this paper a new CNN architecture to binary classify text reviews into positive and negative, then, we have applied multiple deep learning optimizers in our CNN so as to determine the best and relevant optimization algorithm for a such classification. Also, we have trained our model on three different datasets and we have examined the performance of our model with the optimizers using the accuracy metric.

The contributions of our paper are like so: i) A new CNN architecture for a binary classification of text reviews; ii) Our CNN model reaches a good accuracy and great performance against the state-of-the-art models; and iii) The adaptative optimizers algorithms perform better using the new CNN in text reviews classification compared to other optimizers.

The remain of the paper is arranged such as; i) Section 2 displays a review of some corresponded studies that employed CNN models for TC problem; ii) In section 3, we proposed our deep learning CNN architecture, the datasets, and some model settings plus the implemented optimizers; iii) Section 4 illustrates the experimental results; and iv) Lastly, we conclude our paper and we present some perspectives.

2. RELATED WORK

In this section we explore some state-of-the-art studies which utilized deep learning algorithms and CNN models for TC purposes. Actually, CNNs become a common and an efficient model architecture for TC problems [13]. Over the years, many researchers proposed different CNN-based models aiming to extract features from text and predict the intended labels. Kalchbrenner *et al.* [14] have suggested a model called dynamic CNN (DCNN). As shows Figure 1, the DCNN involves an embedding layer that builds a sentence matrix for every word in a given sentence. Then, the wide convolutional layers and the dynamic pooling layers map over the sentence to produce a feature relation between the words in the sentence. In practice, the dynamic k-max-pooling parameter takes value based on the sentence length and the position of the convolutional layer.

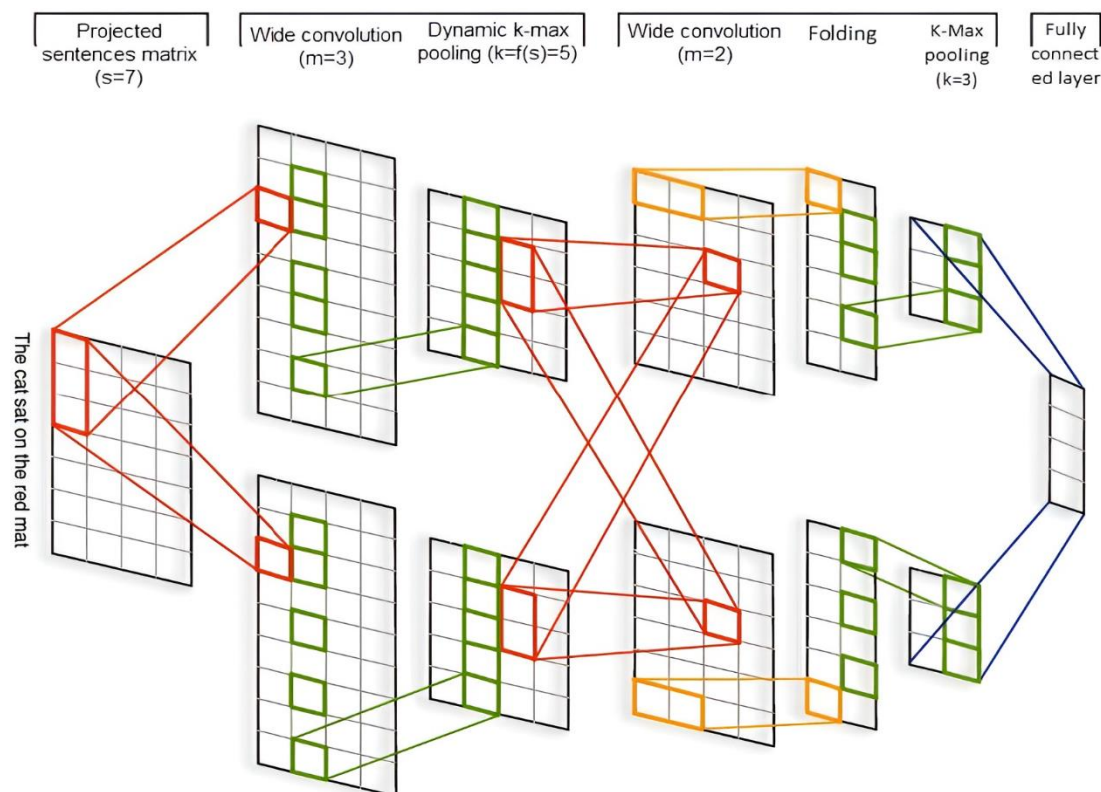


Figure 1. The DCNN architecture [14]

Next, Kim [15] have presented a light CNN architecture as illustrated in Figure 2, based on one convolutional layer and filters for TC problem. Actually, the Kim's model contains an embedding layer, a

convolutional and a max pooling layer, followed by a fully connected layer with dropout, plus a softmax output. Effectively, the author has used the unsupervised embedding model word2vec, and he has compared four initialization approaches to learn the word embeddings. All the Kim's approaches have enhanced the researches in TC and sentiment analysis problems with CNN [13].

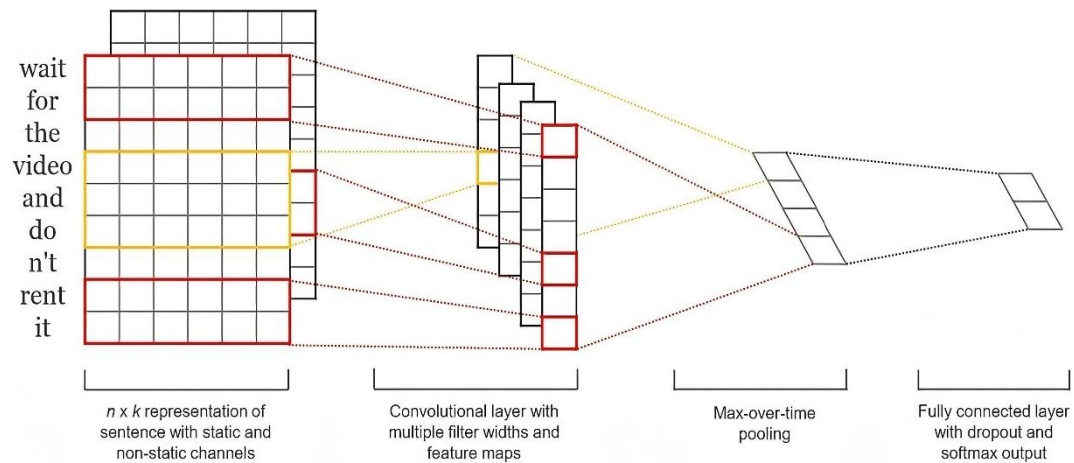


Figure 2. The Kim's CNN architecture [15]

In fact, there have been many attempts to improve the Kim's architecture. Johnson and Zhang [16] have trained the embedding of small parts of text using an unlabeled text data, then the embeddings have been fed to the CNN model as labeled data for TC. As well, the authors have suggested a deep pyramid CNN (DPCNN) [17] which included a deep neural network to increase the computational complexity and also its performance. Therefore, Liao *et al.* [18] have converted the input sentences into matrices, then each sentence matrix has been represented by a word vector which forms the embeddings for the CNN architecture. The proposed CNN was able to understand sentiments from the tweets. Afterwards, [8], [19] have improved the architecture of CNN model by using consecutive convolutional layers and they have reached good accuracies for sentiment classification. Besides, [20] and [21] have examined different CNN settings to find the optimal CNN configuration and to improve the performance for TC. On the other hand, several recent studies [10], [11], [22] have merged CNN with LSTM for TC purposes. Actually, the authors have fed the convolutional layers of CNN with word embeddings, then the output has been appended to the LSTM layers in order to learn long-term dependencies between words. Finally, the softmax layer takes the output from the LSTM layers and produces the classification result.

3. RESEARCH METHOD

3.1. The proposed network

Our suggested CNN model as described in Figure 3, employed two convolutional layers and a max pooling layer, plus two fully connected layers. Actually, we started tokenizing our train data through a vocabulary file which that contains the most frequent words. Then, we randomly initialize the embedding layer to extract meaningful features from the train process. In practice, the embedding layer received the input words and produced feature values for them. Later, each word will be regrouped standing on the learned meaning. Afterward, the two convolutional layers took the output from the embedding layer, slid a window using a kernel size, and applied filters for every window in order to collect more features. Indeed, we appended a dropout layer to each convolutional layer so as to ignore non optimal features. Next, the max-pooling layer selected the maximum values from the convolutional layers and provided this output to two fully-connected layers. In effect, we applied a dropout layer to the first fully-connected layer intending to avoid overfitting, and an activation function rectified linear unit (ReLU) for the exponential growth in computation. Later, the second fully-connected layer produced the vector result which involves a positive or negative classification value. Finally, a Softmax function predicted the label result based on a probability calculation to each class. In regards to the optimization, we applied the binary cross entropy as a loss function, then we compare a set of optimizers with our suggested CNN to identify the best implemented models. The results of our architecture network with several optimizer algorithms are presented in the next section.

3.2. Experiments

Our experiments were performed with Python and TensorFlow framework in Google Colab notebook using Google compute engine backend, central processing unit (CPU) mode and 12.68 GB of memory. Actually, we implemented our CNN model to binary classify reviews into positive and negative from three popular datasets in TC: Amazon reviews [23], internet movie database (IMDb) movie-reviews [24], [25], and rotten tomatoes movie-reviews data [24], [26]. In addition, we experimented a set of optimizers with our CNN model to determine, using empirical examination, the best optimizer for TC and sentiment analysis problems.

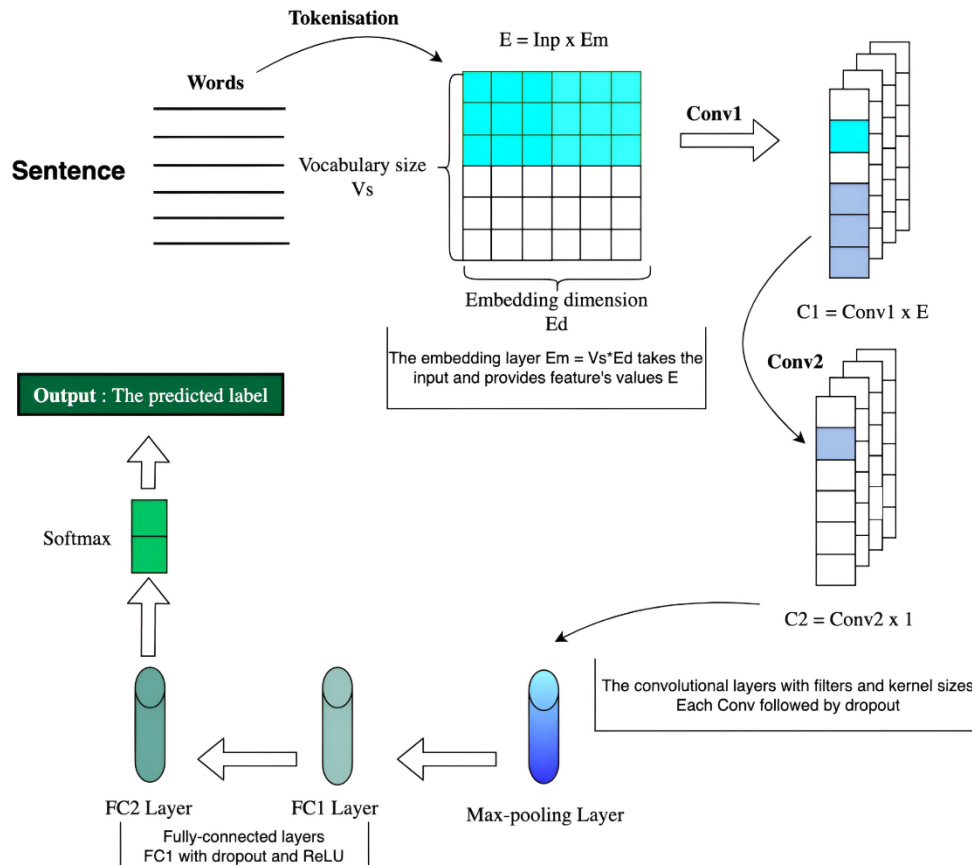


Figure 3. The proposed CNN model for text classification

As follows, we describe the implemented optimizer algorithms:

- Gradient descent (GD) [27]: is the renowned optimization algorithm employed to perform neural networks. GD utilizes calculus and adjusts the values consistently so as to attain the local minimum. It computes the gradient of the cost function standing on the count of the dataset. In effect, there are three types of GD: (1) Batch gradient descent which computes the gradient of the cost function for the complete dataset, (2) stochastic gradient descent (SGD) which generates parameters adaptation to every training sample, and (3) the Mini-batch gradient descent that updates the parameters for each mini-batch.
- Momentum [28]: Actually, using the SGD takes a noisy and more steeply path than GD because of changing parameters in each training example, which means a slow computation time to reach the optimal minimum. Hence, the momentum algorithm surpasses this problem by appending a fragment of a previous oscillation update to the current oscillation, so the process accelerates the time steps and becomes faster.
- Adagrad [29]: is a gradient-based optimizer that adapts the learning rate standing on frequent and infrequent parameters. The more the parameters change, the less the learning rate gets updates. Otherwise, it generates a little update of the learning rate for frequent parameters and a large update for the infrequent ones. Therefore, it is widely used in case of a sparse data training.

- AdagradDA [29]: refers to adagrad dual averaging which is an adagrad-based algorithm. This optimizer adjusts the regularization of unseen features on each mini batch. Indeed, AdagradDA is basically applied for large sparsity in linear models.
- Adadelta [30], [31]: is an advanced algorithm of Adagrad optimizer that adjusts the decaying of the learning rate whereby the model could learn more features. In practice, the algorithm utilizes variables to fix the size of some accumulated gradients.
- FTRL [32]: “Follow the (Proximally) regularized leader” is a GD-based algorithm with an alternative representation of the L1 regularization and model coefficients. The optimizer uses a per-coordinate learning rates; besides, it has a high sparsity and convergence properties.
- Root mean squared propagation (RMSprop) [33]: is an unpublished optimizer suggested in coursera class by Geoff Hinton [33]. The optimizer stands on an adaptive learning rate method. Similar to Adadelta, RMSprop reduces the monotonically decreasing learning rate and accelerates the optimization. In effect, the algorithm utilizes an average of squared gradients that decays exponentially for dividing the learning rate.
- Adam [34]: or adaptive moment estimation is an optimizer that employs adaptive learning rates to update every network weights parameter. Actually, Adam is an alternative extension of SGD and also inherits features from Adagrad and RMSprop. Effectively, it requires fewer parameters tuning and lower memory requirements. Furthermore, it is widely used to solve non-convex problems with large datasets in a faster running time.

Next, we present some parameter values we used in our CNN model,

- For embedding layer, we employed an embedding dimension $Ed=150$ with a sequence length of $S=500$ and a maximum size of vocabulary words of $vocab_size=5,000$.
- Concerning convolutional layers, we set the kernels sizes to the following; $k1=5$, $k2=3$ and the number of filters to $F1=256$, and $F2=128$.
- In connected layers, we defined the count of hidden layers as $H1=128$, $H2=164$.

For the optimization, we set a dropout to 0.5 and a learning rate with $1e-3$ to initialize the whole optimizer algorithms.

3.3. Datasets

In the current section, we proffer some information on the implemented data. Actually, we applied all our CNN models on three text reviews datasets with different sizes. As shown in Table 1, we have classified the datasets into large, medium, and small regarding the number of reviews. More details on datasets are given:

- Amazon reviews [23] contains 4,000,000 customers’ reviews up to March 2013 about several product categories. Besides, the data is labeled into two classes depending on the review scores ratings from 1 to 5. ‘Positive’ is represented by 5 and 4 stars, and ‘Negative’ by 1 and 2 stars. For experiments, we employed 100,000 reviews from the data which represents the large dataset type.
- Rotten Tomatoes (RT) reviews dataset [24], [26] includes 5331 positive snippets of text RT movie-reviews and 5331 negatives ones. RT was first utilized in Pang/Lee ACL 2005 [26] and it is a medium dataset in comparison with the previous one.
- IMDB reviews [24], [25] is about 2,000 sentiment text reviews regarding movies. The data contains 1,000 negative and 1000 positive sentences introduced in Pang/Lee ACL 2004 [25]. The IMDB movie-reviews is considered as a small dataset.

Table 1. Number of reviews and types of data in the three datasets

Data	Type	Number of reviews
Amazon reviews [23]	Large dataset	100,000
RT reviews [24]	Medium dataset	10,662
IMDB reviews [24]	Small dataset	2,000

Practically, we split each data-type into three sets: train, validation and test. Then, we pre-processed our train data using several text filters in order to remove noisy contents. Afterwards, we build from each input sentence the label and the content to train our models. The results of each model are represented and described in the next section.

4. RESULTS AND DISCUSSION

In the current section, we describe the obtained results using different optimizer algorithms with our CNN architecture. As shown in Table 2, we applied each one of the optimizer models on three types of datasets:

Optimizer algorithms and convolutional neural networks for text classification (Mohammed Qorich)

large, medium and small to explore the impact of the data amount on the optimizer's efficiency. In effect, we evaluated the efficiency of the optimizer's models by the accuracy metric.

For readability, we entitled the models using the selected optimizer for the CNN architecture. For example, CNN-Gradient-descent represents the model that employed the Gradient descent as an optimizer in the CNN model. As a loss function, we utilized the cross entropy in the whole models. Actually, the results illustrate that the best CNN model for the small and medium-data is CNN-Adam. However, the CNN-RMSprop has surpassed the CNN-Adam model in the large-dataset, despite the good accuracy reached by CNN-Adam.

Table 2. The results of the optimizer models for each dataset-type

Model	Accuracy (%)		
	Small-dataset	Medium-dataset	Large-dataset
CNN-Gradient-descent	50	50.75	52.85
CNN-Momentum	50	54.03	66.33
CNN- Adagrad	51.5	51.31	50.69
CNN- AdagradDA	51	50	50
CNN-Adadelata	50	50	50.46
CNN-FTRL	50	50	50
CNN-RMSprop	60.75	71.15	90.48
CNN-Adam	70.5	74.58	90.01

In practice, we notice that the accuracy in the CNN-Gradient-descent model is poor and changed the values by little steps, which means that the model learned slowly even if the data amount gets larger. Otherwise, with CNN-Momentum model, the accuracy increased and the model obtained better performance. In effect, the momentum method accelerates GD in the pertinent directions and reduces oscillations. On the other hand, the other optimizers have attained inadequate performances and their accuracy kept a stable value in the three types of datasets. Meanwhile, the RMS-prop model achieved a good accuracy in the large-data and overall, it performed better than the other models in the small and medium-datasets. Actually, the RMS-prop converges faster and requires less parameters tuning than GD algorithms and their variants. Incidentally, the CNN-Adam displayed its advancement opposing all the other optimizers and it achieved great efficiencies regardless the data amount. In fact, the Adam optimizer takes advantages from various optimizer algorithms and overpasses the other optimizers in term of computation time, parameter requirements and ease of implementation.

Figures 4 and 5 show the validation accuracy of the three datasets for our best performed models; CNN-Adam and CNN-RMSprop. Actually, we notice that the more the data get larger, the more the accuracy increases and achieves good performance. Also, the two optimizers with our CNN architecture have made a well progress and a good curve trace in the first 100 epochs.

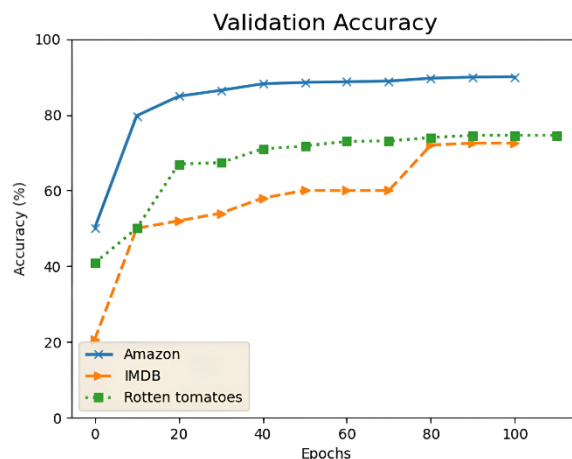


Figure 4. The CNN-Adam validation accuracy

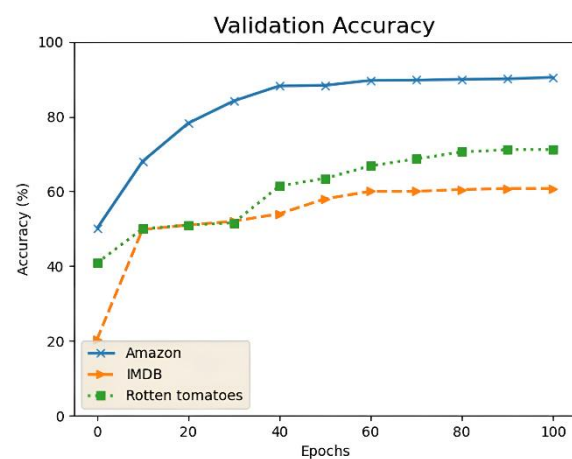


Figure 5. The CNN-RMSprop validation accuracy

On the other hand, Table 3 makes a comparison between our best CNN model result and other state of the art CNN models, using the accuracy metric. Actually, the CNN-RMSprop model which implements the RMSprop optimizer with our CNN architecture has obtained the best efficiency with 90.48% accuracy. This

model classified the text reviews as negative or positive from the Amazon-reviews dataset, and surpassed the most CNN binary classification models from the state-of-the-art.

Table 3. Comparison of the results of our best CNN model and some related works models

Model	Accuracy (%)
Chen's and Wang's CNN [22]	76.09
Kim's and Jeong's CNN [8]	81.06
Kim's CNN [15]	81.5
Johnson's and Zhang's CNN [16]	85.7
Feng's and Cheng's CNN [9]	86.32
DCNN [14]	86.8
Rehman's <i>et al.</i> CNN [11]	87
Jain's <i>et al.</i> CNN [10]	87.1
CNN-RMSprop (our model)	90.48

5. CONCLUSION AND PERSPECTIVES

In this paper, we suggest a new CNN architecture to binary classify text reviews as negative or positive. Our suggested model has examined a set of optimizer algorithms to evaluate empirically the best optimizer for sentiment analysis problem. The experiments had shown that RMSprop and Adam are the most efficient models. Moreover, we obtained great performances compared with the mentioned state of the art architectures by reaching an accuracy of 90.48%. As a perspective, we plan to implement our model for a multi-classification text problem and promote the architecture performances.




REFERENCES

- [1] Q. Li *et al.*, "A survey on text classification: From shallow to deep learning," 2020, [Online]. Available: <http://arxiv.org/abs/2008.00364>.
- [2] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, "Survey on text classification algorithms: From text to predictions," *Information (Switzerland)*, vol. 13, no. 2, 2022, doi: 10.3390/info13020083.
- [3] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics (Switzerland)*, vol. 9, no. 3, 2020, doi: 10.3390/electronics9030483.
- [4] S. Pal, S. Ghosh, and A. Nag, "Sentiment analysis in the light of LSTM recurrent neural networks," *International Journal of Synthetic Emotions*, vol. 9, no. 1, pp. 33–39, 2018, doi: 10.4018/ijse.2018010103.
- [5] A. Chamekh, M. Mahfoudh, and G. Forestier, "Sentiment analysis based on deep learning in E-commerce," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13369 LNAI, pp. 498–507, 2022, doi: 10.1007/978-3-031-10986-7_40.
- [6] S. Sachin, A. Tripathi, N. Mahajan, S. Aggarwal, and P. Nagrath, "Sentiment analysis using gated recurrent neural networks," *SN Computer Science*, vol. 1, no. 2, 2020, doi: 10.1007/s42979-020-0076-y.
- [7] M. Zulfarnain, R. Ghazali, M. Aamir, and Y. M. M. Hassim, "An efficient two-state GRU based on feature attention mechanism for sentiment analysis," *Multimedia Tools and Applications*, 2022, doi: 10.1007/s11042-022-13339-4.
- [8] H. Kim and Y. S. Jeong, "Sentiment classification using Convolutional Neural Networks," *Applied Sciences (Switzerland)*, vol. 9, no. 11, 2019, doi: 10.3390/app9112347.
- [9] Y. Feng and Y. Cheng, "Short text sentiment analysis based on multi-channel CNN with multi-head attention mechanism," *IEEE Access*, vol. 9, pp. 19854–19863, 2021, doi: 10.1109/ACCESS.2021.3054521.
- [10] P. K. Jain, V. Saravanan, and R. Pamula, "A hybrid CNN-LSTM: A deep learning approach for consumer sentiment analysis using qualitative user-generated contents," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 20, no. 5, 2021, doi: 10.1145/3457206.
- [11] A. U. Rehman, A. K. Malik, B. Raza, and W. Ali, "A hybrid CNN-LSTM model for improving accuracy of movie reviews sentiment analysis," *Multimedia Tools and Applications*, vol. 78, no. 18, pp. 26597–26613, 2019, doi: 10.1007/s11042-019-07788-7.
- [12] S. M. Zaman, M. M. Hasan, R. I. Sakline, D. Das, and M. A. Alam, "A comparative analysis of optimizers in recurrent neural networks for text classification," *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering, CSDE 2021*, 2021, doi: 10.1109/CSDE53843.2021.9718394.
- [13] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep learning based text classification: A comprehensive review," 2020, [Online]. Available: <http://arxiv.org/abs/2004.03705>.
- [14] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, vol. 1, pp. 655–665, 2014, doi: 10.3115/v1/p14-1062.
- [15] Y. Kim, "Convolutional neural networks for sentence classification," *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1746–1751, 2014, doi: 10.3115/v1/d14-1181.
- [16] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *Advances in Neural Information Processing Systems*, vol. 2015-January, pp. 919–927, 2015.
- [17] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization," *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, vol. 1, pp. 562–570, 2017, doi: 10.18653/v1/P17-1052.
- [18] S. Liao, J. Wang, R. Yu, K. Sato, and Z. Cheng, "CNN for situations understanding based on sentiment analysis of twitter data," *Procedia Computer Science*, vol. 111, pp. 376–381, 2017, doi: 10.1016/j.procs.2017.06.037.
- [19] B. M. Mulyo and D. H. Widyanoro, "Aspect-based sentiment analysis approach with CNN," *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, vol. 2018-October, pp. 142–147, 2018, doi:




- 10.1109/EECS.2018.8752857.
- [20] M. Pota, M. Eposito, G. De Pietro, and H. Fujita, "Best practices of convolutional neural networks for question classification," *Applied Sciences (Switzerland)*, vol. 10, no. 14, 2020, doi: 10.3390/app10144710.
 - [21] M. A. Nasichuddin, T. B. Adji, and W. Widyawan, "Performance improvement using CNN for sentiment analysis," *IJITEE (International Journal of Information Technology and Electrical Engineering)*, vol. 2, no. 1, 2018, doi: 10.22146/ijitee.36642.
 - [22] N. Chen and P. Wang, "Advanced combined LSTM-CNN model for Twitter sentiment analysis," *Proceedings of 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2018*, pp. 684–687, 2019, doi: 10.1109/CCIS.2018.8691381.
 - [23] "Amazon 2013," pp. 1–23, 2016, [Online]. Available: <https://www.kaggle.com/bittlingmayer/amazonreviews>.
 - [24] "IMDB 2004 / Rotten Tomatoes 2005," pp. 1–23, 2016, [Online]. Available: <https://www.cs.cornell.edu/people/pabo/movie-review-data/>.
 - [25] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," *In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pp. 271–278, 2004, doi: 10.48550/arXiv.cs/0409058.
 - [26] B. Pang and L. Lee, "Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales," *ACL-05 - 43rd Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 115–124, 2004, doi: 10.3115/1219840.1219855.
 - [27] N. Ketkar, "Stochastic Gradient Descent," *Deep Learning with Python*, pp. 113–132, 2017.
 - [28] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999, doi: 10.1016/S0893-6080(98)00116-6.
 - [29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *COLT 2010 - The 23rd Conference on Learning Theory*, pp. 257–269, 2010.
 - [30] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, [Online]. Available: <http://arxiv.org/abs/1212.5701>.
 - [31] Y. Wang, J. Liu, J. Misić, V. B. Misić, S. Lv, and X. Chang, "Assessing optimizer impact on DNN model sensitivity to adversarial examples," *IEEE Access*, vol. 7, pp. 152766–152776, 2019, doi: 10.1109/ACCESS.2019.2948658.
 - [32] H. B. McMahan *et al.*, "Ad click prediction: A view from the trenches," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F128815, pp. 1222–1230, 2013, doi: 10.1145/2487575.2488200.
 - [33] N. Shi, D. Li, M. Hong, and R. Sun, "RMS prop parameter converges with proper hyperparameter," *Int. Conf. Learn. Represent.*, vol. 1, no. 2018, pp. 1–10, 2021.
 - [34] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

BIOGRAPHIES OF AUTHORS



Mohammed Qorich    was born in Meknes, Morocco, in 1993. He received the T.U.D degree in Computer Engineering by the High School of Technology of Meknes, Moulay Ismail University (Morocco), in 2013. He received the P.L degree in IT development by the faculty of sciences Ain Chock Casablanca, Hassan II University (Morocco), in 2014. He received the M.S degree in educational technology at the higher normal school of Tetouan, Abdelmalek Essaadi University (Morocco) in 2020. Currently, He is Ph.D. candidate at Moulay Ismail University, Meknes, Morocco. His research interests include natural language processing, deep learning, text classification, and Chatbot. He can be contacted at email: mohamedqorich@gmail.com.



Rajae El Ouazzani    received her master's degree in Computer Science and Telecommunication by the Mohammed V University of Rabat (Morocco) in 2006 and the Ph.D. in Image and Video Processing by the High National School of Computer Science and Systems Analysis (Morocco) in 2010. From 2011, she is a Professor in the High School of Technology of Meknes, Moulay Ismail University in Morocco. Since 2007, she is an author of several papers in international journals and conferences. Her domains of interest include multimedia data processing and telecommunications. She can be contacted at email: elouazzanirajae@gmail.com.